

Version 2024b

Grundlagen der Web-Programmierung

Übungsaufgaben

Prof. Dr.-Ing. Tenshi Hara
tenshi.hara@ba-sachsen.de



HTML

AUFGABE 1

Erstellen Sie ein Beispieldokument für eine Untermenge von HTML namens μ HTML, wobei die möglichen Tags und Attribute beschränkt sind auf:

- ohne Attribute: `<html>`, `<head>`, `<title>`, `<style>`, `<body>`
- mit Attributen `id` und `class`: `<p>`, ``, ``

AUFGABE 2

Diskutieren Sie mit Ihren Kommilitonen Kernunterschiede zwischen den einzelnen HTML-Versionen: HTML 4.01, XHTML 1.0, HTML 5!

AUFGABE 3

Erstellen Sie eine „Hello World“-Seite konform zu HTML 4.01.

AUFGABE 4

Erstellen Sie eine „Hello World“-Seite konform zu XHTML.

AUFGABE 5

Ergänzen Sie Ihre „Hello World“-Seiten um

- Überschriften unterschiedlicher Ebenen,
- Absätze,
- Trennlinien und
- Texte mit Sonderzeichen, insbesondere Umlaute und ß.

AUFGABE 6

Machen Sie Ihre „Hello World“-Seiten explizit kompatibel zu UTF-8.

AUFGABE 7

Machen Sie die „Hello World“-Seiten nun explizit inkompatibel zu UTF-8.

AUFGABE 8

Ergänzen Sie Ihre HTML-4-Hello-World-Seite nun um Anker und ein Inhaltsverzeichnis am Anfang der Seite!

Die Elemente des Inhaltsverzeichnisses sollen anklickbar sein und zu den jeweiligen Ankern verlinken.

AUFGABE 9

Erstellen Sie ein neues Dokument, welches zwei echte Frames (frame-Tag verwenden!) nebeneinander beinhaltet, die jeweils 50% der Seitenbreite einnehmen.

- Im linken Frame binden Sie Ihre HTML-4-„Hello World“-Seite ein.
- Im rechten Frame binden Sie Ihre XHTML-„Hello World“-Seite ein.

AUFGABE 10

Erzeugen Sie den gleichen Effekt, den Sie in Aufgabe 9 erzielt haben mit zwei nebeneinander liegenden Iframes!

Hinweis: Sie werden etwas CSS benötigen.

CSS

AUFGABE 11

Erstellen Sie ein neues HTML-4-Dokument mit diversen Beispieltexten, Containern, Textbereichen (``), Absätzen, Überschriften und Ankern.

- Weisen Sie den Containern und Textbereichen Klassen und IDs zu.
- Erstellen Sie ein CSS-Dokument zur optischen Gestaltung der zuvor definierten Klassen und IDs.
 - Nutzen Sie Angaben zu Schriftart, Schriftgröße, Schriftfarbe, Schrifttyp (kursiv, fett, unterstrichen), Umrahmung, Hintergrundfarbe, etc.!
 - Binden Sie das CSS-Dokument in das HTML-Dokument ein

AUFGABE 12

Ändern Sie die Eigenschaften der Anker und Links aus Aufgabe 11 in der Art, dass ein Mouse-Over-Effekt entsteht (bspw. Ändern der Farbe, der Schriftdicke, etc.).

AUFGABE 13

Fügen Sie ein Bild in das Dokument aus Aufgabe 12 ein.

- Definieren Sie Bildgröße und Position mittels CSS!
- Ergänzen Sie das Bild um Mouse-Over-Effekte.

AUFGABE 14

Fügen Sie zwei weitere Bilder in das Dokument aus Aufgabe 13 ein.

Geben Sie jedem der nun drei Bilder unterschiedliche Positionstypen (statisch, fixiert, relativ) und prüfen Sie Anzeige durch Skalierung des Browser-Fensters und Scrollen.

AUFGABE 15

Ändern Sie die CSS-Eigenschaften Ihres Dokumentes aus Aufgabe 14 so, dass es unterschiedliche Darstellungsangaben für Bildschirm- und Druck-Ausgabe enthält.

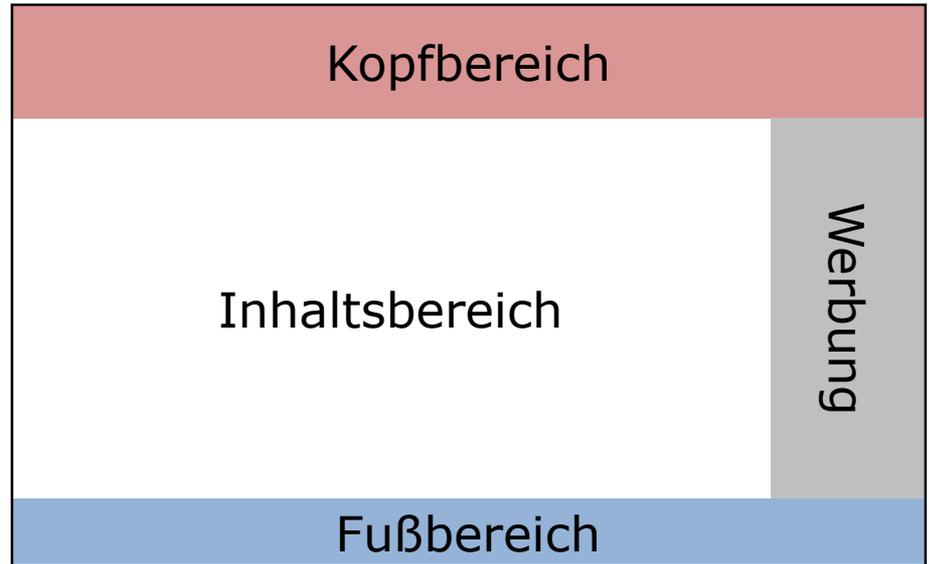
Prüfen Sie mit Hilfe der Druckvorschau Ihres Browsers, ob der gewünschte Effekt erreicht wird. → **Drucken Sie nicht tatsächlich!** Die armen Bäume!

AUFGABE 16

Erstellen Sie eine Web-Seite mit einer Reihe verlinkter Dokumente:

- 1 Startseite
- 1 Impressum
- 1 Kontaktseite
- 1 Galerie
- 5 (unterschiedliche) Produkte

Jede Seite soll nebenstehendem Gestaltungsmuster entsprechen!



- Kopfbereich, Fußbereich und Werbung sollen nicht mitscrollen!
- Im Kopfbereich liegen die Navigationselemente.
- Im Fußbereich befinden sich Links zum Kontakt und zum Impressum
- Kopf- und Fußbereich liegen auf oberster Ebene und überdecken alles andere (lassen Sie im Inhaltsbereich genügend Platz, damit kein Inhalt durch diese Bereiche überdeckt wird!)

AUFGABE 17

Unabhängig davon, wie Sie die Web-Seite aus Aufgabe 16 umgesetzt haben, erstellen Sie nun insgesamt drei Versionen der Web-Seite:

1. mit **Frames**, damit Sie Elemente wie den Kopf- und den Fußbereich wiederverwenden können,
2. mit **Iframes**, damit Sie Elemente wiederverwenden können, aber auch flexibler bei der Gestaltung sind, und
3. mit **div**-Containern, damit die maximale Flexibilität bei der Gestaltung haben. (ggf. können Sie zum Wiederverwenden von Elemente Iframes in die Container legen).

Speichern Sie die Version mit div-Containern als Ausgangspunkt für spätere Aufgaben! (im Folgenden als „**Template-Seite**“ bezeichnet)

JavaScript

AUFGABE 18

Ergänzen Sie die Template-Seite um eine Begrüßungsmeldung, die Besuchern angezeigt wird sobald die Startseite lädt. Die Meldung soll dabei als Alert umgesetzt sein.

AUFGABE 19

Ergänzen Sie den Alert aus Aufgabe 18 um mindestens einen erzwungenen Zeilenumbruch.

AUFGABE 20

Ergänzen Sie den Alert aus Aufgabe 19 in der Art, dass einige berechenbare Ausdrücke enthalten sind, z.B.

- aktuelles Datum und aktuelle Uhrzeit
- Meta-Informationen zum Browser

AUFGABE 21

Ergänzen Sie den Alert aus Aufgabe 20 in der Art, dass weitere berechenbare Ausdrücke enthalten sind, z.B.

- Fenstergröße
- Position eines Objektes
- das Ergebnis einer mathematischen Berechnung (Wurzel, o.ä.)

Stellen Sie sicher, dass der JavaScript-Kode blockierungsfrei arbeitet indem Sie Browser-abhängige Funktionen in `try...catch`-Blöcke legen.

AUFGABE 22

Ändern Sie die Template-Seite nun in der Art, dass der Alert aus Aufgabe 21 nicht mehr automatisch angezeigt wird.

Stattdessen soll der Alert durch das Anklicken einer Schaltfläche auf der Startseite ausgelöst werden.

AUFGABE 23

Ergänzen Sie den Code aus Aufgabe 22 in der Art, dass die Schaltfläche nach dem Anklicken aus dem DOM entfernt wird.

AUFGABE 24

Ergänzen Sie den Code aus Aufgabe 23 in der Art, dass zusätzlich eine neue Schaltfläche erzeugt wird, mit der Sie die erste Schaltfläche wieder herstellen können. Dabei soll dann die zweite Schaltfläche ihrerseits entfernt werden.

Hinweis: Statt des Entfernens aus dem DOM können Sie auch die CSS-Eigenschaften in der Art ändern, dass die Schaltfläche nicht mehr angezeigt werden.

AUFGABE 25

Ergänzen Sie die Template-Seite um eine Schaltfläche zum Einloggen.

Nach Betätigung der Schaltfläche soll ein Cookie gesetzt und anschließend die Seite neu geladen werden.

Ist das entsprechende Cookie gesetzt, soll statt der Einloggen-Schaltfläche eine zum Ausloggen angezeigt werden.

Nach Betätigung der Ausloggen-Schaltfläche soll das Cookie gelöscht und anschließend die Seite neu geladen werden.

CSS3

AUFGABE 26

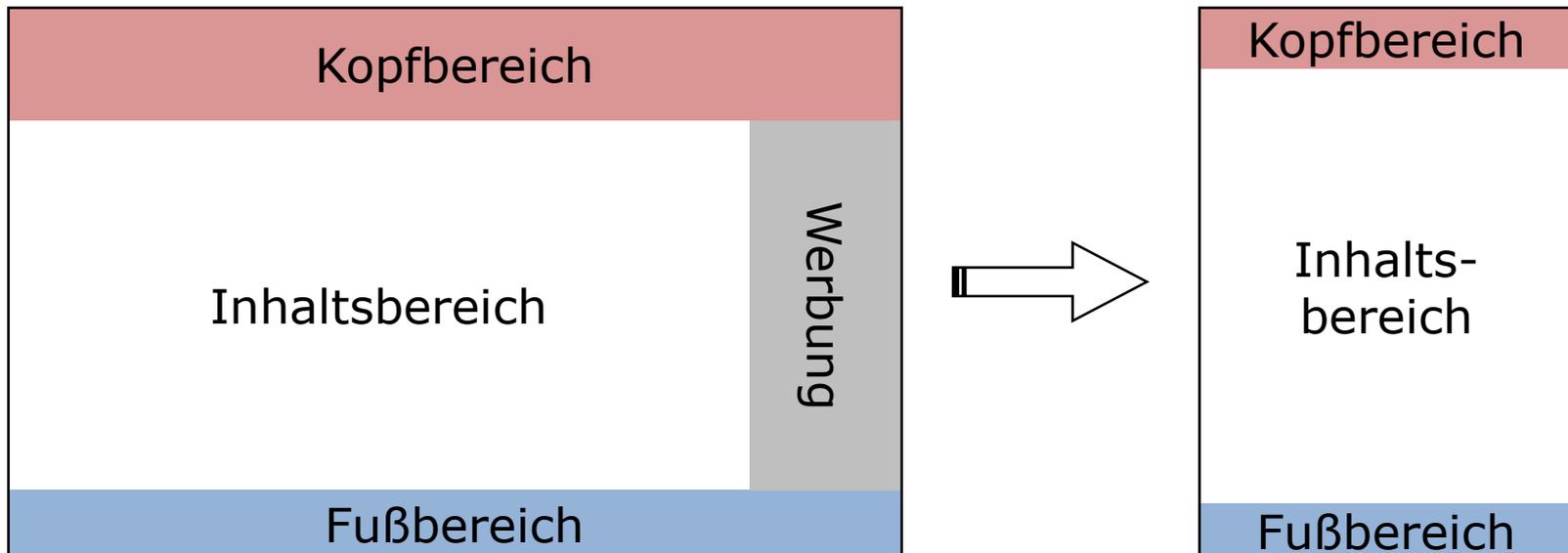
Ändern Sie die Template-Seite so um, dass die Basisschriftgröße im Inhaltbereich mindestens 0,7cm Höhe aufweist.

Alle anderen Schriftgrößen, insbesondere von Überschriften, sollen abhängig von der Basisschriftgröße skalieren.

AUFGABE 27

Ändern Sie die Template-Seite so mit CSS Media Queries um, dass sie verschiedene Bereichsgrößenverhältnisse für unterschiedliche Endgerätetypen und Bildschirmgrößen bzw. -ausrichtungen unterstützt.

Kopf- und Fußbereich sollen dabei als zwingend erforderlich gelten, während der Werbebereich unterhalb einer Mindestbreite wegfallen kann.



AUFGABE 28

Ändern Sie die Template-Seite in der Art, dass die Größe der Kopf- und Fußbereiche sich in Abhängigkeit von der Scroll-Position ändern.

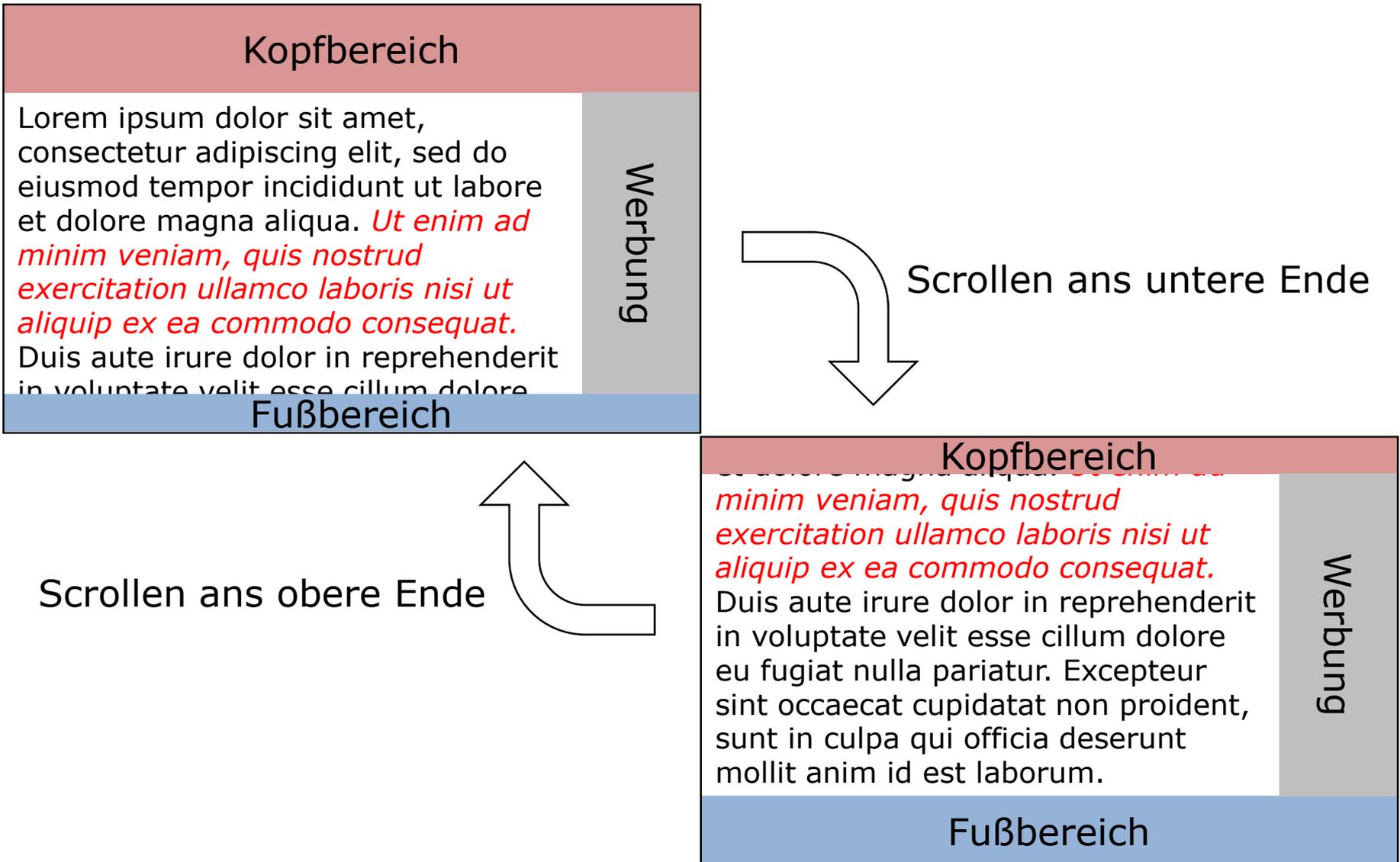
Nutzen Sie dazu

a) eine Kombination von JavaScript und CSS

b) nur die Möglichkeiten von CSS3, z.B. die Positionsangabe `sticky`

- <https://gedd.ski/post/position-sticky/>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/position>
- https://www.w3schools.com/howto/howto_css_sticky_element.asp
- <https://css-tricks.com/scroll-fix-content/>

AUFGABE 28 – BEISPIEL



AUFGABE 29

Ändern Sie die Template-Seite nun in der Art, dass diese mit CSS Media Queries die Geräte-Layout-Muster

- Column Drop bzw.
- Layout Shifter

für den Inhaltsbereich umsetzt.

Speichern Sie die Version mit dem Layout Shifter als Ausgangspunkt für spätere Aufgaben! (im Folgenden als „Master Template“ bezeichnet)

Falls noch nicht geschehen, sollten Sie spätestens jetzt die Ergebnisse von Aufgabe 25 in das Master Template übernehmen.

AUFGABE 30

Binden Sie eine kostenlose Web-Font in das Master Template ein!

AUFGABE 31

Ergänzen Sie zwei neue CSS-Klassen, mit denen Sie die Schaltfläche zum Ein-/Ausloggen abhängig vom Login-Zustand unterschiedlich dekorieren.

Die Dekoration soll per JavaScript zugewiesen werden.

AUFGABE 32

Ergänzen Sie den Kopfbereich um eine Personalisierungsauswahl. Dabei soll der Nutzer aus einem Drop-Down-Menü aus zwei Gestaltungsmustern auswählen können welche

- die Farbgestaltung,
 - die Schriftart und
 - die Schriftgröße
- ändern.

Dabei sollen die Varianten wie folgt umgesetzt sein:

- Normal: Serifenschrift mindestens 12pt, kräftige Farben
- Barrierefrei: serifenlose Schriftart, mindestens 14pt, hoher Kontrast

HTML5

AUFGABE 33

Realisieren Sie die Umsetzung von Aufgabe 25 (Einloggen/Ausloggen mit Cookie) durch die Nutzung des HTML5 LocalStorage statt durch Cookies.

AUFGABE 34

Realisieren Sie die Umsetzung von Aufgabe 25 bzw. Aufgabe 33 (Einloggen/Ausloggen mit Cookie) durch die Nutzung des HTML5 SessionStorage statt durch Cookies.

Vergleichen Sie nun die Lösungen der Aufgaben 33 und 34 hinsichtlich der Persistierung des Login-Zustands beim Schließen des Tabs bzw. Fensters!

AUFGABE 35

Erzeugen Sie im Master Template zwei neue Unterseiten für Medieninhalte.

Binden Sie auf der ersten Seite ein kurzes Video ein, wobei unterschiedliche Codecs angeboten werden sollen. Vor dem Abspielen soll ein Symbolfoto angezeigt werde.

Binden Sie auf der zweiten Seite eine kurze Tonspur ein. Diese soll beim Laden der Webseite automatisch abgespielt werden und in Schleife laufen.

Freie Medien bspw. unter <https://lern.es/3MI-GLWEB-20/Medien/>.

AJAX

AUFGABE 36

Lesen Sie aufmerksam die Informationen auf

https://www.w3schools.com/xml/ajax_xmlhttprequest_response.asp

Ergänzen Sie anschließend das Master Template um eine neue Unterseite zum Testen von AJAX.

Die Seite soll ein Formular enthalten über das ein kurzer Text an den Server gesendet werden kann.

Des Weiteren soll ein div-Container in der Nähe des Formulars die Antwort des Servers anzeigen.

Nutzen Sie zum Testen das folgende Server-Skript*:

<http://ajax.lern.es/ajax36.php>

* Beachten Sie, dass das Skript keine HTTPS-Unterstützung hat, damit Sie lokal testen können. Wegen SOP und CORS ist ansonsten kein Zugriff von file:/// auf https:// möglich.

AUFGABE 37

Ergänzen Sie das Dokument aus Aufgabe 36 um ein weiteres Formular. In diesem Formular soll Auto-Complete ab dem vierten eingegebenen Zeichen ermöglicht werden.

Beachten Sie, dass Sie die Rückgabewerte des Servers (JSON) konvertieren müssen!

Die Auswahl eines Eintrags aus der zurückgegebenen Liste soll das Textfeld entsprechend ausfüllen.

Nutzen Sie zum Testen das folgende Server-Skript*:
<http://ajax.lern.es/ajax37.php>

* Beachten Sie, dass das Skript keine HTTPs-Unterstützung hat, damit Sie lokal testen können. Wegen SOP und CORS ist ansonsten kein Zugriff von file:/// auf https:// möglich.

Aufgabe 37 — AJAX-Demo

Wort:

- Watvogels
- Wundervogel
- Zugvogel
- vogelaehnlichen
- Vogel
- Vogelfreund
- Vogelmensch

AUFGABE 38

Ändern Sie das Dokument aus Aufgabe 37 in der Art ab, dass Sie im Auto-Complete den Match hervorheben können.

Nutzen Sie dazu einen regulären Ausdruck!

(Die performantere Lösung via `str.replace` nutzen Sie **nicht!**)

Wenn Sie eine Auswahl treffen (in der Ergebnisliste einen Eintrag anklicken), soll die Auswahl in das Suchfeld eingefügt werden.

Hinweis: Nutzen Sie reguläre Ausdrücke, um im Dropdown-Menü das Word-Match optisch hervorzuheben.

Aufgabe 38 — AJAX-Demo

Wort:

- Galgenvogel
- Galgenvogels
- Kanarienvogel
- Lockvogel
- Papiervogel**
- Pechvogel
- Raubvogel

Prüfungsaufgabe

PRÜFUNGSAUFGABE

Für die Prüfungsaufgabe ist das Aussehen und die Funktionalität des finalen Master-Template als Zielsetzung definiert.

Prüfungsaufgabe:

Ersetzen Sie an geeigneten Stellen Ihren Code durch neuen Code eines JavaScript-Toolkits Ihrer Wahl (z.B. pojo, jQuery, ...). Stellen Sie sicher, dass die AJAX-Komponente funktional vollständig ist.